Detección de Outliers Aplicando Algoritmo de Optimización basado en el Apareo de Abejas

Jorge L. Rodriguez

e-mail: cereal333@gmail.com

and

Ramón D. Sandoval

e-mail: santiagosan23@yahoo.com

and

Gustavo D. Pacheco

e-mail: gustavodavidp@gmail.com

Departamento de Investigación Operativa, Universidad Nacional de Salta.

Abstract

This work has the objective of adding a proposition for the detection of outliers. That is to say, to detect data belonging to a sample that possess extreme values that the difference of the rest and allow the investigator to suspect about the origin of the same ones [1]. As for the methodology used in this work, it has been opted by a heuristic method, since it is sought to reach the objective before signal through the picked up theoretical and empiric study of the literature it has more than enough heuristic algorithms [2]. In this proposal we present a heuristic one well-known as Honey Bee Mating Optimization Algorithm, with the purpose of finding feasible solutions efficiently. The experimental results on different datasets of the literature, demonstrate the quality of our algorithm compared with the heuristic algorithm based on local search (LSA) of Zengyou He [3].

Keywords: Outlier, Entropy, Metaheuristic, HBMO, Data Mining.

Resumen

Este trabajo tiene el objetivo de sumar una propuesta para la detección de casos atípicos. Es decir, detectar datos pertenecientes a una muestra que posean valores extremos que los diferencia del resto y permitan al investigador sospechar acerca del origen de los mismos [1]. En cuanto a la metodología utilizada en este trabajo, se ha optado por un método heurístico, ya que se pretende alcanzar el objetivo antes señalado a través del estudio teórico y empírico recogido de la literatura sobre algoritmos heurísticos [2]. En esta propuesta presentamos una heurística conocido como Algoritmo de Optimización basado en el Apareo de Abejas, con el propósito de encontrar soluciones factibles eficazmente. Los resultados experimentales sobre distintos datasets de la literatura, demuestran la calidad de nuestro algoritmo comparado con el algoritmo heurístico basado en búsqueda local (LSA) de Zengyou He [3].

Palabras Claves: Outlier, Entropía, Metaheurística, HBMO, Minería de Datos.

1. INTRODUCCIÓN

La Minería de Datos ofrece un rango de técnicas que permiten identificar casos atípicos basados en modelos. Estos modelos se pueden clasificar en:

1. Modelos de datos inusuales. Con este modelo se pretende detectar comportamientos raros en un dato respecto a su grupo de comparación o con el mismo, por ejemplo la consignación de altas sumas de dinero en efectivo. Para este caso se puede emplear técnicas de análisis de clutering seguido de un análisis de detección de outliers.

2. Modelos de relaciones inexplicables. Con este modelo se pretende encontrar relaciones de registros que tienen iguales valores para determinados campos, resaltando el hecho que la coincidencia de valores debe ser auténticamente inesperado, desechando similitudes obvias como el sexo, la nacionalidad, por ejemplo la transferencia de fondos entre dos o más compañías con la misma dirección de envío. Para este caso se pueden aplicar técnicas de Clustering para encontrar grupos sospechosos y reglas de asociación.

3. Modelos de características generales. Con este modelo se pretende una vez detectados ciertos casos, hacer predicciones de futuros ingresos de transacciones sospechosas. Para estas predicciones usualmente se emplean técnicas de regresión, árboles de decisión y redes neuronales.

De igual forma, taxonómicamente la minería de datos se puede dividir en dos clases: descriptiva y predictiva [4], [5] como se presenta en la **figura 1**.

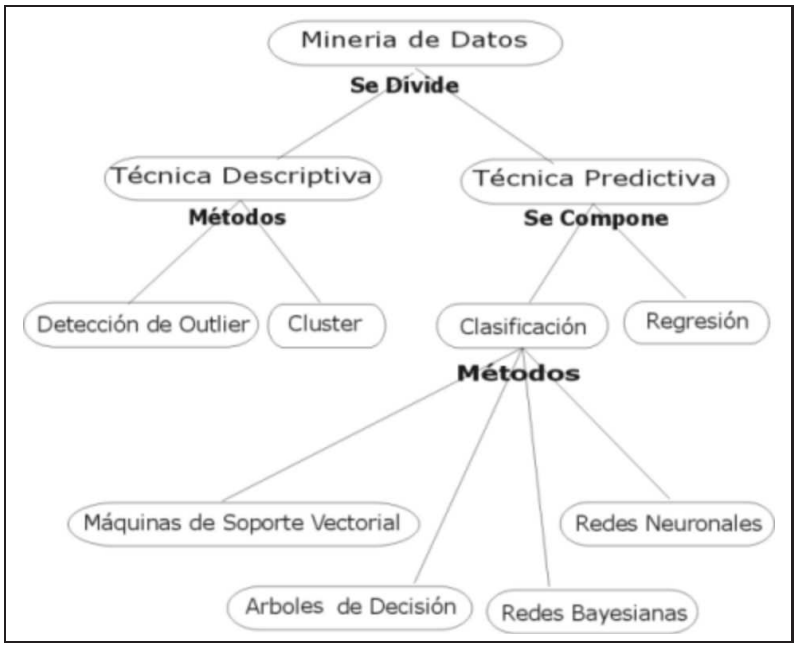

Figura 1: Representación taxonómicamente de la minería de datos

¿Por qué es importante detectar los valores atípicos? Fundamentalmente, por sus consecuencias [6]:

- Distorsionan los resultados al oscurecer el patrón de comportamiento del resto de casos y obtener conclusiones que, sin ellos, serían completamente distintas.

- Puede afectar gravemente a una de las condiciones de aplicabilidad mas habituales de las técnicas multivariantes, la normalidad.

Una definición [1] declara que un valor atípico es una observación que se desvía tanto de otras observaciones, que despierta la sospecha de que se generó por un mecanismo diferente.
Para proteger los resultados de estas posibles observaciones atípicas, se aplican procedimientos denominados métodos de detección de *outliers*. Estos métodos se basan en ciertas *medidas* que nos permiten averiguar si una observación es atípica; siendo candidata a ser estudiada a fondo.
En la mayoría de estos procedimientos se utilizan como elementos ciertos estimadores de posición y escala, de manera que las buenas propiedades de dichos procedimientos dependen de los comportamientos de estos estimadores. Por esta razón, a los estimadores utilizados en los métodos de detección de outliers se les pide ciertas propiedades en presencia de observaciones atípicas, en definitiva que sean robustos.
Desde un punto de vista sistemático, un dataset que contiene muchos outliers tiene una gran cuota de desorden; en otros términos, quitando los outliers del conjunto de datos se obtiene un dataset menos desordenado. Basado en esta observación, el problema de outliers podría definirse informalmente como un problema de optimización de la siguiente manera:

Encontrar un subconjunto pequeño del dataset designado, tal que el grado de desorden del dataset resultante se minimice después de extraer este subconjunto.

La Entropía en la teoría de información es una buena opción por medir el grado de desorden de un dataset. Nosotros apuntaremos a minimizar la entropía, es decir, apuntamos a encontrar k outliers del dataset original donde k es el número esperado de outliers en el conjunto de datos.
Hasta ahora, el problema de optimización podría describirse de una manera más concisa:

Encontrar un subconjunto de k objetos, tal que la entropía del dataset se minimice después de quitar este subconjunto de objetos.
A continuacion se muestra en la **Figura 2** la representación gráfica de la instancia Iris, donde se puede visualizar de forma clara puntos que no respetan el comportamiento general de los datos, a estos puntos se los denomina *outliers*.

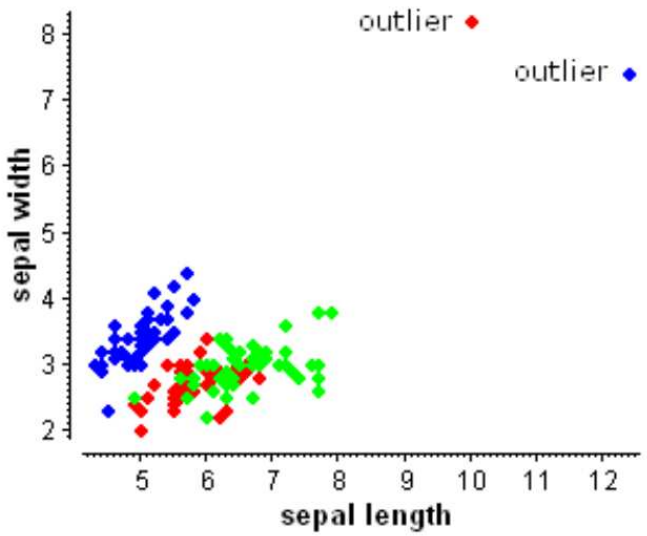

Figura 2: Representación gráfica de la instancia Iris

El estudio realizado comienza en la *Sección 3*, presentando en términos estadístico-matemático la Entropía, como así también el procedimiento para detectar outliers. En la *Sección 4*, presentamos la solución propuesta. En la *Sección 5*, se muestran las pruebas realizadas con nuestro algoritmo y sus respectivos resultados. Por último, en la *Sección 6* realizamos un comentario final a modo de resumen.

2. EL PROBLEMA

En esta sección, presentamos aspectos sobre la entropía, y de forma consecuente formulamos el problema.

2.1. Entropía

La Entropía es la medida de información e incertidumbre de una variable azarosa [7]. Formalmente, si X es una variable aleatoria, S (X) el conjunto de valores que X puede tomar, y p (x) la función de probabilidad de X, la Entropía E(X) se define como se muestra en la Ecuación (1).

$$E(X) = - \sum_{x_. \epsilon S(x)} p(x) log(p(x)) \quad (1)$$

La entropía de un vector multidimensional $x = [X_1, ..., X_m]$ puede computarse tal como se muestra en la Ecuación (2).

$$E(x) = - \sum_{x_{1.} \epsilon S(x_1)} ... \sum_{x_{m.} \epsilon S(x_m)} p(x_1, ..., x_m) log(p(x_1, ..., x_m)) \quad (2)$$

2.2. Formulación del Problema

El problema que estamos intentando resolver puede formularse como sigue. Dado D un dataset de n puntos p_1 ,..., p_n, donde cada punto es un vector multidimensional de m atributos, por ejemplo, $\vec{p_i}$= (p_i1 ,..., $p_i m$) y dado un valor k entero, se desea obtener un subconjunto $O \subseteq D$ de tamaño k, de tal manera de minimizar la entropía de D - O. Es decir:

$$\textbf{\textit{min}}\ E_{O \subseteq D}(\text{D - O }) \quad (3)$$

$$\text{sujeto a } \mid O \mid = k$$

En este problema, necesitamos calcular la entropía de un conjunto de registros usando la Ecuación (2). Para hacer el cálculo más eficaz, efectuamos una simplificación en el cálculo de la entropía. Asumiendo la independencia de los registros, transformamos la Ecuación (2) en la Ecuación (4). Es decir, la probabilidad de los atributos combinado con los valores, da como resultado el producto de las probabilidades de cada atributo, entonces la entropía pueden calcularse como la suma de las entropías de los atributos [3].

$$E(\text{X}) = - \sum_{x_{1.} \epsilon S(x_1)} ... \sum_{x_{m.} \epsilon S(x_m)} p(x_1, ..., x_m) log(p(x_1, ..., x_m))$$

$$= E(X_1) + E(X_2) + ... + E(X_n) \quad (4)$$

2.3. Otros procedimientos para resolver el Problema

En la literatura se pueden encontrar diferentes procedimientos para resolver el problema. Uno de esos casos se expone en el trabajo de Ortega Dato [8]. A continuación un resumen del mismo: El procedimiento propuesto es una medida, denominada distancia por truncamientos, basada en las $\alpha Truncadas$ y $\alpha\beta Truncadas$, definidas a su vez mediante las conocidas medias truncadas. Esta distancia, para cada observación de la muestra, nos informa de lo apropiado que es suponer que dicha observación es una realización del experimento que se está estudiando.

Otro trabajo que se puede mencionar es de Atkinson Gordo A. D. J., Ariza López F. J. y García-Balboa J. L. [9]. Aquí se presenta una herramienta capaz de solventar el problema de detección de outliers: los estimadores robustos. Éstos permiten ponderar valores que se encuentran más alejados de los centrales en una serie. Así, se analiza algunos de los principales estimadores M de Huber, método Danés, Geman y McClure, y se aplican a la detección y ponderación de valores atípicos en un control de exactitud posicional planimétrico sobre poblaciones sintéticas contaminadas artificialmente.

3. SOLUCIÓN PROPUESTA

En esta sección, presentamos HBMO (algoritmo de optimización basado en el apareo de abejas) [10], el cual ha resultado ser eficaz y eficiente en la identificación de outliers.

3.1. HBMO

A continuación presentamos una idea general de HBMO. Primero daremos conceptos básicos para su correcta interpretación.

- Una colonia de abejas está compuesta usualmente de una abeja reina (queen), miles de zánganos (drones) y abejas obreras (workers).
- Una colonia puede tener una o varias abejas reinas.
- Las abejas reinas depositan los huevos.
- Una abeja reina vive 5-6 años, los zánganos y las obreras viven no más de 6 meses.
- La única tarea de los zánganos es aparearse con la abeja reina. Una vez realizado el apareo muere.
- Las obreras se encargan de cuidar la colonia y a veces ponen huevos.

En el proceso de apareo, la reina sale del panal seguida por los zánganos. Durante el vuelo, la abeja reina se aparea con los zánganos. En cada apareo el esperma de los zánganos se acumula para formar el pool genético (espermateca) de la colonia.
Al comienzo del vuelo, la abeja reina sale del panal con cierta energía o velocidad, y a medida que ocurre el apareo su velocidad o energía disminuye. Esto se debe al volumen del esperma recolectado. El apareo se realiza en forma probabilística similar a Simulated Annealing. Esto quiere decir que no todos los zánganos llegan a aparearse.
La probabilidad de apareo se puede calcular de la siguiente manera:

$$prob(\text{reina,zángano}) = \exp\,[-\Delta(f)/S(t)]$$

donde:

$\Delta(f)$ es el valor absoluto entre la función objetivo del zángano y la función objetivo de la reina.

S(t) es la velocidad de la reina en el tiempo t

Con estas aclaraciones de puede decir que la probabilidad de apareo es alta:

si

la reina recién comienza su vuelo

ó

la $f$obj (zángano) $\approx$ $f$obj (reina)

Cada vez que pasa el tiempo: S(t+1) = α(t) * S(t), donde α(t) es un valor entre [0,1] la velocidad o energía de la reina disminuye.

La función de las obreras es mejorar el genotipo de los huevos fertilizados. Esta función puede representarse como un conjunto de heurísticas. Es decir, las obreras (heurística) perturban y mejoran la solución. Mientras mas obreras se propongan mejores soluciones se obtendrá.

La lista de zánganos se obtiene a través de la generación de soluciones aleatorias. La actualización de dicha lista se realiza a través de soluciones no usadas (zánganos que no se aparearon) y de generaciones aleatorias (nuevos zánganos).

Las obreras mejoran las nuevas soluciones obtenidas de los apareos de la reina. De esta nueva generación se elige la mejor solución, luego se remplaza a la actual reina por la reina de la nueva generación. Esto se realiza tantas veces hasta que la nueva reina o nueva solución no mejore.

A continuación, en la **Figura 3** se muestra un esquema de HBMO.

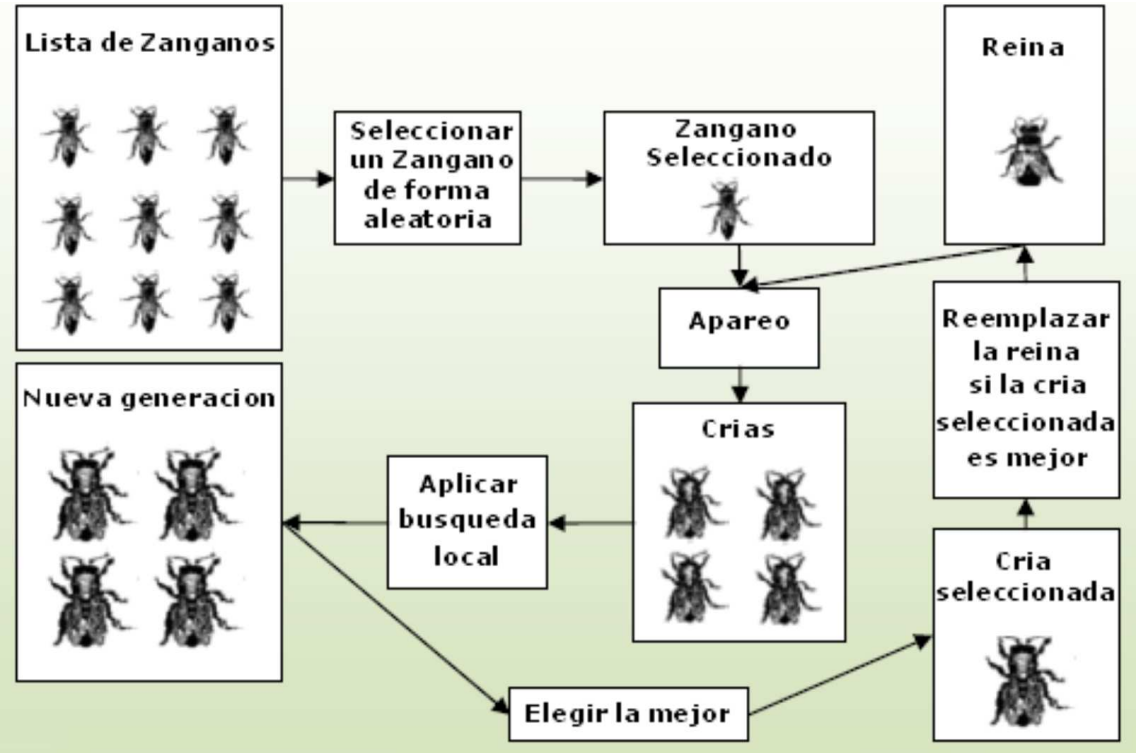

Figura 3: Algoritmo HBMO

3.2. Cruzamiento

Con respecto al apareo, se puede realizar por dos métodos, *Crossover* y *mutatión*.
El proceso de crossover se produce al mezclar información genética de dos individuos, tal como la reproducción sexual, obteniéndose un tercer individuo que comparte la información de ambos. A continuación presentamos los operadores usados:

- Crossover de punto simple

  Este operador indica el punto de corte, es decir, dado un cromosoma de largo N se busca un punto de corte en forma aleatoria que determina las porciones que compondrán el nuevo individuo. A continuación se muestra el proceso donde se han considerado dos individuos con 11 variables binarias cada uno:

  Padre 1 0 1 1 1 0 0 1 1 0 1 0
  Padre 2 1 0 1 0 1 1 0 0 1 0 1

  Luego se selecciona el punto de corte, por ejemplo en la quinta posición, y se aplica el crossover, de donde se crean dos nuevos individuos:

  Hijo 1 0 1 1 1 0 | 1 0 0 1 0 1
  Hijo 2 1 0 1 0 1 | 0 1 1 0 1 0

- Crossover de dos puntos

  El crossover de dos puntos es similar al caso anterior, pero en este se seleccionan dos puntos de corte en forma aleatoria, sin repetir los puntos y en orden creciente. De esta forma, las variables entre los sucesivos puntos de crossover son intercambiados entre los dos padres creándose dos nuevos hijos. Es importante notar que la sección ante el primer corte no cambia para ninguno de los individuos. A continuación, se muestra el proceso.

  Padre 1 0 1 1 1 0 0 1 1 0 1 0
  Padre 2 1 0 1 0 1 1 0 0 1 0 1

  La posiciones de corte son: 2 y 6

  Hijo 1 0 1 | 1 0 1 1 | 1 1 0 1 0
  Hijo 2 1 0 | 1 1 0 0 | 0 0 1 0 1

- Crossover uniforme

  Este operador generaliza este esquema haciendo de cualquier posición un lugar potencial para efectuar el crossover. Se crea entonces una mascara, cuyo largo es igual al de un individuo y su estructura aleatoria. Luego se aplica la mascara a los padres, donde la paridad de bits indica como se obtendrá el hijo. Considerando dos individuos de 11 variables binarias cada uno se tiene:

Padre 1 0 1 1 1 0 0 1 1 0 1 0
Padre 2 1 0 1 0 1 1 0 0 1 0 1

Cada variable que conforma los nuevos individuos es seleccionada aleatoriamente, y con igualdad de probabilidades a partir de los padres. A modo de ejemplo, se tiene que el hijo 1 se forma tomando un bit del padre 1 si el correspondiente bit de la mascara es 1 o bien del padre 2 si el bit de la máscara es cero.
El hijo segundo es creado usando el sistema inverso.

Mascara 1 0 1 1 0 0 0 1 1 0 1 0
Mascara 2 1 0 0 1 1 1 0 0 1 0 1

Una vez aplicado el operador los nuevos individuos son:

Hijo 1 1 1 1 0 1 1 1 1 1 1 1 1 1
Hijo 2 0 0 1 1 0 0 0 0 0 0 0 0 0

3.3. Algoritmo basado en HBMO

En la **Figura 4** se muestra el seudocódigo del Algoritmo HBMO para la detección de outliers.

HBMO (k, Z, e, V_{max}, V_{min}, α, TC, D)

k Cantidad de outliers
z Número de zanganos
e Capacidad de espermateca

V_{max} Velocidad de la reina al inicio de un vuelo de apareamiento
V_{min} Velocidad de la reina al final de un vuelo de apareamiento
α Tasa de reducción de velocidad

TC Tipo de Crossover a utilizar
D Dataset

Inicio
Generar un conjunto P de posibles outliers
Generar aleatoriamente $\alpha \epsilon$ (0, 1)
espermateca = [ ]
$Condicion = True$
Mientras Condicion
Generar aleatoriamente Z zánganos de k elementos cada uno, desde D
Elegir el mejor zángano como reina (R)
Generar aleatoriamente $V_{max} \epsilon$ [0.5, 1]
Generar aleatoriamente $V_{min} \epsilon$ [0, V_{max})
Mientras $|espermateca| <$ e **y** $V_{max} > V_{min}$
Seleccionar aleatoriamente un zángano Z_i
Calcular $\Delta(f) = |f(R) - f(Z_i)|$ /* f: función de entropía /*
Generar aleatoriamente t ϵ [0,1]
Si $exp\ (-\Delta(f)\ /\ Vmax) >$ t
Aplicar Crossover TC entre Z_i y R
Agregar los 2 nuevos individuos a *espermateca*
Sino $V_{max} = V_{max} * \alpha$
Elegir la mejor cria C desde *espermateca*
Para cada elemento del conjunto P
Intercambiar P_i con algún C_j que haga decrecer
lo mas posible la Entropía de C
Si $f(C) >= f(R)$ /* f: función de entropía /*
$Condicion = false$
Mostrar C, f(C)
Fin

Figura 4: Algoritmo HBMO propuesto

4. PRUEBAS

En esta sección, se incluye información sobre las instancias usadas en las pruebas, como así también los valores de los parámetros usados en el algoritmos y los resultados obtenidos.

4.1. Hardware Usado

El Hardware que se usó para realizar las pruebas computacionales es:

- CPU = Intel Pentium D 2.8Ghz dual core.
- Memoria RAM = 256MB (DDR400).

4.2. Parámetros del Algoritmo

Se listan los parámetros del algoritmo y los valores aplicados durante la etapa de prueba.

- Numero de reinas = 1
- Cantidad de zánganos = 100
- Tamaño del espermateca = 20
- Factor de reducción $\alpha = 0.98$
- Tipo de cruzamiento = 2 cortes
- $V_{max} \epsilon [0.5, 1]$
- $V_{min} \epsilon [0, V_{max})$

4.3. Resultados Obtenidos

A continuación se muestran los resultados obtenidos sobre cada una de las tres instancias utilizadas. Los resultados se resumen en una tabla que muestra:

- Cantidad de outliers que se pretende encontrar.
- La entropía obtenida.
- Tiempo promedio que lleva encontrar la cantidad de outliers pretendida.
- Error relativo (entre entropía alcanzada y entropía de la solución obtenida por LSA).

Tabla 1: Resultados obtenidos para 10 iteraciones usando la instancia Lymphography

Cant.Outlier	Algoritmo HBMO		Algoritmo LSA		Error (%)
	Entropía	Tiempo (seg)	Entropía	Tiempo (seg)	
3	21,835268	6,64200	21,835268	13,14100	0
5	21,456307	8,20300	21,456307	23,01399	0
10	20,790636	14,37399	20,790636	53,48600	0
15	20,361850	23,60899	20,361850	134,70199	0
20	19,977226	35,78199	19,977226	201,79700	0
25	19,717631	51,01600	19,721349	273,10900	0,018853
30	19,498626	78,90499	19,484692	380,62499	0,071513

Tabla 2: Resultados obtenidos para 10 iteraciones usando la instancia Iris

Cant.Outlier	Algoritmo HBMO		Algoritmo LSA		Error (%)
	Entropía	Tiempo (seg)	Entropía	Tiempo (seg)	
3	19,361678	09,40500	19,361678	19,54600	0
5	19,268267	12,15599	19,268267	33,90500	0
10	19,042126	21,22000	19,042144	111,86099	0,000095
15	18,832208	39,23600	18,821876	164,43600	0,054894
20	18,610555	56,20199	18,597348	261,82799	0,071016
25	18,401272	69,62599	18,400644	292,15499	0,003413
30	18,206688	98,59300	18,196712	440,21900	0,054823

Tabla 3: Resultados obtenidos para 10 iteraciones usando la instancia Pima

Cant.Outlier	Algoritmo HBMO		Algoritmo LSA		Error (%)
	Entropía	Tiempo (seg)	Entropía	Tiempo (seg)	
3	46,601238	99,43800	46,601238	1003,40699	0
5	46,556909	124,39000	46,556909	1742,90699	0
10	46,449054	209,01599	46,449054	3884,15799	0
15	46,346978	322,31299	46,346822	6351,32799	0,000337
20	46,251178	463,75000	46,248981	9148,97000	0,004750
25	46,158234	659,78199	46,153317	12529,56300	0,010654
30	46,061481	976,93599	46,060292	15755,14999	0,002581

A medida que las pruebas se fueron realizando notamos que la entropía de nuestro algoritmo se aproxima a la entropía del algoritmo exhaustivo LSA [3].
Con respecto al tiempo computacional, la diferencia es muy marcada debido a que el algoritmo LSA abarca casi todas las combinaciones posibles de resultado.

5. CONCLUSIONES

En resumen, nuestro algoritmo detecta la presencia de outliers en forma eficiente y permite un ahorro evidente de tiempo computacional, comparado con el algoritmo LSA.
Por otra parte, los outliers detectados y estimados con nuestro algoritmo corresponden, en su mayoría, a las conclusiones alcanzadas con el algoritmo LSA.

Como trabajo futuro, se espera analizar otras instancias de prueba, como así también aplicar mejores operadores de coeficiente e intentar mejorar los resultados obtenidos.
Finalmente, se aplicará el algoritmo sobre datos de casos reales como la detección de *spam*.

Referencias

[1] Hawkins Douglas M. Identification of outliers. Chapman and Hall, Reading, London, p:128-135, 1980.

[2] Mohammand Fathian, Babak Amiri, Ali Morosi. Aplication of Honey Bee Mating Optimization on Clusterig, p:1502–1513, 2007. FALTA PUBLICACION

[3] Zengyou He, Xiaofei Xu, Shengchun Deng. An Optimization Model for Outlier Detection in Categorical Data, p:400-409, 2005.

[7] Shannon C.E.. Amathematical theory of communication. Bell System Technical Journal, p:379-423, 1948.

[4] Jiawei Han. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2006.

[5] Oded Maimon and Lior Rokach. Data Mining and Knowledge Discovery Handbook, p:1-17, Speinger 2005. VERIFICAR

[6] Joaquín Aldás Manzano, Ezequiel Uriel Jiménez. Analisis Multivariante Aplicado: Aplicaciones al marketing, p:22, 2006. FALTA PUBLICACION

[8] Ortega Dato. Detección de observaciones atípicas mediante truncamientos, p:47-68, 2002.

[9] Atkinson Gordo A. D. J., Ariza López F. J., García-Balboa, J. L. Estimadores robustos: una solución en la utilización de valores atípicos para el control de la calidad posicional. GeoFocus (Artículos), nº 7, p:171-187, 2007.

[10] Bozorg Haddad o., Afshar A.. Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization, p.661-680, 2006.